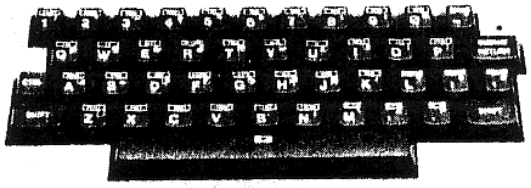
VeeZed MICRO MAGIC 4

BEGINNER'S GUIDE TO BASIC



MICRODATA

The stock in trade of any computer is its memory. Fortunately, although the VeeZed's internal memory is small, we can still store a limitless amount of data on cassettes. We've already seen how words can be tape-filed for future reference. Programs that allow us to file lists (inventories) are known as DATABASE programs.

MICRODATA is a VeeZed Database. Like the pages of a book, your monitor screen will display your choice of 40 pages. Each page has space for 10 entries. That's 400 items of information!!! And that's not all. MICRODATA allows you to manipulate your entries. You can INSERT an item and have the existing entries all move up to make room for it. You can DELETE an entry and the rest will close up to fill the gap. Any number of entries you choose can be rearranged into ALPHABETICAL ORDER, and, of course, you can TAPE-FILE the pages of your choice for future reference. A page, or pages can be printed out on a plotter or a printer. There is even provision for giving the file a name.

```
10  ' ***********************
20  '*     VEEZED    * MICRO  *
30  '*  MICRODATA    * MAGIC  *
40  ' ***********************
```

```
100 CLS:CLEAR1000:DIMZ$(400):COLOR3,1
```

CLS to clear the screen.
CLEAR1000 reserves 1K of memory.
DIMZ$(400) will allow us to use String Variable Z$ with 400 subscriptions. Z$(1) will be the first entry on Page 1 and Z$(400) will be the last entry on Page 40.
COLOR - you can vary this to suit yourself.

```
110 GOSUB940:PRINT@36,"**MICRODATA**"
```

*(* = SHIFT J GRAPHIC)*

GOSUB940 - this subroutine will be entered next.
PRINT program title on screen.

```
940 FORJ=1TO400:Z$(J)="-":NEXT:FORJ=1TO400STEP10:Z$(J)="*":NEXT
950 RETURN
```

There are 400 places where data may be entered. When the program is first loaded all these places are empty. The first LOOP puts a HYPHEN into the first place of every line. Later HYPHENS will be used to indicate an empty line.
The second LOOP then puts an ASTERISK in first place of the first line of each page, in place of the HYPHEN. The ASTERISK will be used to indicate a blank page.
You will see later how these can be used to prevent empty lines and empty pages using up unnecessary time and space.

```
120 PRINT@100,"ENTER FILE NAME":PRINT:PRINT"    ";:INPUTF$
130 IFF$=""THENF$="NO NAME"
```

UNDERLINE = INVERSE.
____ = 4 SPACES.
String Variable F$ is used to print the file name.
If no name is entered F$ will simply print "NO NAME".

```
140 CLS:PRINT@67,"MENU␣␣";F$:PRINT
150 PRINT"␣␣␣␣CATALOGUE":PRINT"␣␣␣PAGE":PRINT"␣␣␣␣LOAD"
160 PRINT"␣␣␣␣SAVE":PRINT"␣␣␣␣HARD COPY":PRINT"␣␣␣␣NEW FILE"
170 PRINT"␣␣␣␣ALPHABETICAL SORT":PRINT"␣␣␣␣ENTER INDEX LETTER"
```

This is the MENU.
PRINT means Print on screen.
HARD COPY means Print on paper.
␣␣␣␣ = spaces. Underline = INVERSE.

MAIN MENU

```
190 A$=INKEY$:A$=""
200 A$=INKEY$:IFA$=""THEN200
210 IFA$="P"THEN390
220 IFA$="H"THEN520
230 IFA$="A"THEN850
240 IFA$="C"THEN280
250 IFA$="S"THEN1070
260 IFA$=L$THEN1000
270 IFA$="N"THENRUNELSE200
```

"L"

```
!!!!!!!!!!!!!!!!!!!!!!!!!!
! DON'T FORGET TO USE   !
! TEMPORARY LINES TO TEST !
!  RUN EACH SUB-ROUTINE  !
!     WHEN ENTERED.      !
!!!!!!!!!!!!!!!!!!!!!!!!!!
```

With A$ set at INKEY$ it will become whatever key is pressed.
If nothing is pressed, or if a wrong key is pressed, the computer will remain at line 200.
NOTE: All the IFs are GOTOs, not GOSUBs.

The first subroutine to be entered will be the CATALOGUE.

CATALOGUE

```
280 CLS:PRINT@3,"CATALOGUE␣␣";F$:PRINT
290 A=1:FORC=1TO40
292 IFZ$(A)="*"THEN350
294 IFZ$(A)="-"THEN310ELSEPRINT"PAGE"C;
300 PRINTZ$(A):GOSUB330
310 A=A+10
320 NEXT:GOSUB340:GOTO400
330 FORT=1TO400:NEXT:SOUND31,1:RETURN
340 A$=INKEY$:A$=""
350 PRINT"END OF CATALOGUE":PRINT"MENU PAGE:SOUND28,2
360 A$=INKEY$:IFA$=""THEN360
370 IFA$="M"THEN140
380 IFA$="P"THEN390ELSE360
```

As you know, subscripted variables can start with a Zero.
In this program the lines of Data should be from Z$(0) to Z$(399). To avoid confusion we decided to beigin with Z$(1) so that they will be identical with the actual data line number.
In line 290 we have made A = 1.
The LOOP (FORC-1TO40) will identify each of the 40 pages.
In lines 292 and 294 the lines are tested to see if they begin with either an ASTERISK or a HYPHEN.
If the first character of a page is a * then the loop jumps to 350 and prints END OF CATALOGUE. This means that the rest of the pages are unused.
If the first character of a page is a - then we jump to 310 where A is increased by 10 so that the next page can be tested, but if there is neither a * or a - the line 294 prints the word PAGE followed by its number.
Line 300 then prints the first entry on that page before hopping off to line 330 for a short pause, a NEXT, a short SOUND and a RETURN.
This RETURN takes us back to line 310 and the C loop continues.
When the LOOP is completed we then get a GOSUB340 which brings in a sort of mini-menu. This is to allow a selection between the MENU or the next PAGE.
You will have gathered that the first entry of each page can be used for a sub-title to be shown on the CATALOGUE index.

The following subroutine selects a page.

```
390 CLS:PRINT@230,"PAGE NUMBER";:INPUTP:IFP<1ORP>40THEN390
400 L=P*10-9:CLS
410 PRINT"PAGE␣"P;"␣␣";F$:PRINT
420 FORI=1TO10:PRINTUSING"###";L;:PRINT"␣";Z$(L):L=L+1
430 NEXT:SOUND25,1
440 PRINT"␣ENTER LINE INSERT":PRINT"␣␣␣␣␣␣␣DELETE PAGE MENU"
450 A$=INKEY$:A$=""
460 A$=INKEY$:IFA$=""THEN460
470 IFA$="L"THEN690
480 IFA$="I"THEN720
490 IFA$="D"THEN780
500 IFA$="P"THEN390
510 IFA$=M"THEN140ELSE460
```

DEBUG AS YOU GO.

SELECT PAGE AND LINE

Line 390 clears the screen and prints "PAGE NUMBER?", the Question Mark being provided by the INPUT. After typing in the required page number it is tested to see if it is less than 1 or greater than 40.
Line 400 calculates the first LINE number from the PAGE number that you have typed in. Variable **L** contains that number. **P** will be used to display the page selected.
Line 410 prints the Page number and the file-name
Line 420 is the loop for printing the 10 lines ready for for you to INSERT information.
Line 430 ends the loop with a short sound.
Line 440 is another mini-menu giving you the choice of selecting a LINE (by number), then INSERT, DELETE, another PAGE or the main MENU. The INKEY$ direct your selection to the appropriate sub-routine.

The following subroutine is for printing a HARD COPY.

```
520 PRINT@417,"********COPY TO PRINTER*******"
530 A$=INKEY$:A$=""
540 PRINT@450,"ENTER PAGE(S)  FILE  MENU":SOUND28,1
550 A$INKEY$:IFA$=""THEN550
590 PRINT"␣␣ENTER START PAGE NUMBER";:INPUTP
600 PRINT"␣␣␣␣␣␣␣␣FINAL PAGE NUMBER";:INPUTF:SOUND28,1
610 X=1:LPRINTF$:LPRINT
620 FORI=PTOF:LPRINT"PAGE"I
630 FORJ=XTOX+9:IFZ$(J)="*"THENLPRINT"␣␣PAGE EMPTY":GOTO140
640 LPRINTUSING"##";J;:LPRINT" ";Z$(J)
650 NEXT
660 X=X+10
670 NEXT
680 LPRINT:GOTO140
```

HARD COPY

**** = SHIFT J Graphics. ␣␣␣␣ = Spaces.
If you select **H** (line 220) the program jumps to this sub-routine. You may then select the page or pages to be LPRINTed or the entire file. If you select PAGE(S) you then enter the the page numbers. If only one page then enter the same number twice.
Line 610 LPRINTs the File name. **X** is made **1** ready for the loop in line 630.
The loop beginning on line 620 will take care of the Page numbers, and the loop beginning on line 630 LPRINTs the lines but considered the file to be empty when a ***** is encountered.
LPRINTUSING puts the line numbers in a neat column.
Line 660 increases **X** by ten to turn the pages.

ENTER TEXT

```
690 PRINT@391,"LINE NUMBER";:INPUTL
700 FORD=417TO447:PRINT@D,CHR$(32):NEXT:SOUND30,1
710 PRINT@417,"TEXT";:INPUTZ$(L):GOTO400
```

Line 690. Look at line 440 - it says ENTER LINE INSERT. Line 690 overprints LINE INSERT to read LINE NUMBER and INPUTL provides the question mark. When the required line number has been entered then line 700 wipes out DELETE PAGE MENU and prints TEST? When your text has been entered the program goes to line 400 which re-prints the page with your text in the appropriate line.

INSERT

```
720 PRINT@391,"LINE NUMBER";:INPUTL
730 FORD=417TO447:PRINT@D,CHR$(32):NEXT:SOUND30,1
740 PRINT@417,"TEXT";:INPUTN$
750 FORJ=400TO-L:Z$(ABS(J))=Z$(ABS(J+1)):NEXT
760 Z$(L)=N$
770 GOSUB1600:GOTO400
```

Similar to the above, except in this subroutine we will be INSERTing text.
When the selected page is on the screen the mini-menu permits INSERT. Pressing I wipes out LINE INSERT and over-prints ENTER LINE NUMBER?
The mini-menu words DELETE PAGE MENU are then wiped out and replaced with TEXT?
Lines750 and 760 adjust the line numbers to allow your text to be INSERTed.
Line 770 says GOSUB1600 which replaces the HYPHENS and ASTERISKS which have been moved by the INSERT.
Finally, GOTO400 to reprint the whole page with your INSERTed text in place.

DELETE

```
780 PRINT@391,"LINE NUMBER";:INPUTL
790 X=400-L
800 FORJ=LTOX
810 Z$(J)=Z$(J+1)
820 NEXT
830 Z$(400)="-"
840 GOSUB1500:GOTO400
```

Similar to the last sub-routine except that this time we DELETE an entry and the rest of the lines of data are moved up to fill the gap. GOSUB1500 replaces all the HYPHENS and ASTERISKS.

ALPHABETICAL SORT

```
850 GOSUB960:PRINT@321,"SORT START LINE";:INPUTS:SOUND15,1
860 PRINT@353,"SORT FINAL LINE";:INPUTF:SOUND17,1:J=0:I=0
870 PRINT@416,"********SORTING STANDBY*********":SOUND19,3
880 IFS<1ORS>400ORF<SORF>400THEN970
890 FORJ-STOF:IFZ$(J)="-"THENZ$(J)=" "
900 K=F-S:FORJ=1TOK
910 FORI-FSSTEP-1
920 IFZ$(I)<Z$(I-1)THENTP$=Z$(I):Z$(I)=Z$(I-1):Z$(I-1)=TP$
930 NEXT:NEXT:GOTO400
```

Line 850 - GOSUB960 prints "MICRODATA"
then comes SORT START LINE? - INPUTS followed by
SORT FINAL LINE? - INPUT F.
J and I are brought to ZEROs and are to be used in LOOPS shortly.
Line 870 is a STANDBY as a long SORT (say, the complete file), could take a few seconds.
Line 890 - Here we get rid of any HYPHENS that could interfere with an Alphabetical Sort. A HYPHEN is ASCII 45 while the letter A is ASCII 65, so in a SORT any HYPHENS would be placed before WORDS.
Lines 900 to 930 is the SORT.
Each line of data is in turn transferred to a temporary String Variable SF$ and compared with all the Z$ lines.
If there were 10 lines to be SORTed then the routine would make 100 comparisons. When you SORT about 10 lines it is done amazingly quickly.

```
960 CLS:PRINT@262,"MICRODATA":RETURN

970 PRINT@416,"*********LINES INCORRECT********":SOUND4,9
980 GOTO400
```

The above are a couple of sub-routines. Line 960 belongs to the GOSUB in line 850. Not an essential sub-routine at all. Lines 970 & 980 is the sub-routine from line 880 which is part of the Alphabetical Sort routine.
The stars ****** are SHIFT J GRAPHICS.

RETRIEVE

```
1000 PRINT@33,"********LOAD FROM TAPE********":COLOR4
1010 PRINT" ********START TAPE NOW********:SOUND28,9:CLS
1020 INPUT#"",F$:PRINTF$:PRINT
1030 FORI=1TO400
1040 INPUT#"",Z$(I):PRINT@448,"xxxxxxxxxx"    (10 spaces)
1045 IFZ$(I)="*"THEN1060
1048 PRINTI;Z$(I)
1050 NEXT
1060 PRINT" **********STOP TAPE**********":SOUND30,9:GOTO140
```

Pretty straight forward. Firstly comes the title LOAD FROM TAPE, then a long sound warning to give you the opportunity to start the tape, and then the File Name is recorded by F$.
The quotes in the loading command are left empty. Unfortunately, the VeeZed will not take a String Variable in here.
The spaces to be printed at 448 are to cover up the word FOUND that appears when retrieving Data from tape.
Line 1045 goes to NEXT if a line begins with *, rejecting empty pages.
Line 1048 prints the line number (I) and the retrieved line.

```
1070 CLS:PRINT@417,"********SAVING TO TAPE********"
1090 A$=INKEY$:A$=""
1100 PRINT@450,"ENTER  FULL FILE OR PART ONLY":SOUND28,1
1110 A$=INKEY$:IFA$=""THEN1110
1120 IFA$="F"THENS-1:F=400:GOTO1180
1130 IFA$="P"THEN1150
1140 IFA$="M"THEN140ELSE1110
1150 PRINT"xxENTER START PAGE NUMBER";:INPUTP
1160 PRINT"xxxxxxxxFINAL PAGE NUMBER";:INPUTF
1170 S=P*10-9:F=F*10
1180 CLS:PRINT"**********RECORD NOW**********":SOUND28,9
1190 PRINT#"",F$:PRINTF$
1200 FORI=STOF
1210 PRINT#"",Z$(I)
1212 IFZ$(I)="*"THEN1230
1214 PRINTUSING"##";I;:PRINT" "Z$(I)
1220 NEXT
1230 PRINT"*********STOP  TAPE*********":SOUND30,9:GOTO140
```

RECORD

**** = SHIFT J Graphics. xxxx = Spaces.
BEWARE! There's no line 1080. Sorry 'bout that.
In response to S in the MENU the program jumps to here.
Fisrt, you get the choice of saving either the FULL FILE or PART ONLY. There is also the option of getting back to the Menu should you get to this part of the program by accident or you change your mind.
If you select PART ONLY then you asked to enter the Starting Page number and the Final Page number. Line 1170 converts the PAGE numbers to LINE numbers.
A long sound gives you the opportunity to begin recording.
The file name is recorded and then the loop records all the lines of data, and prints them on the screen. If you are recordimg a long file you can go and make a cuppa, the long warning in line 1230 will let you know when it's done.
Line 1212 is there to prevent empty pages being recorded.

FINALLY, a couple of sub-routines. They are the subs that replace the HYPHENS and ASTERISKS after they have been disturbed during INSERT and DELETE.

```
1500 G=INT(L/10)*10+10:IFZ$(G)="*"THENZ$(G)="-"
1505 IFZ$(G-9)="-"THENZ$(G)="*"
1510 FORI=GTO400STEP10:IFZ$(I)="*"THENZ$(I)="-"
1520 NEXT:G=G+1
1530 FORI=GTO391STEP10:IFZ$(I)="-"THENZ$(I)="*"
1540 NEXT
1550 RETURN
```

```
1600 G=INT(L/10)*10+11
1610 FORI=GTO400STEP10:IFZ$(I)="-"THENZ$(I)="*"
1620 NEXT:G=G+1
1630 FORI=GTO392STEP10:IFZ$(I)="*"THENZ$(I)="-"
1640 NEXT
1650 RETURN
2000 GOTO140
```

If you have a head for Maths, with a bit of Logic thrown in for good measure, you'll be able to see what these lines are doing. Brielfly, they are checking for HYPHENS and ASTERISKS in the <u>WRONG</u> places and changing them to get them into the <u>RIGHT</u> places.

Following is a GP-100 Printer LLISTing....

```
10  ' ***********************
20  ' *     VEEZED    * MICRO  *
30  ' * MICRODATA  * MAGIC  *
40  ' ***********************

100 CLS:CLEAR1000:DIMZ$(400):COLOR3,1
110 GOSUB940:PRINT@36,"[illegible]"
120 PRINT@100,"ENTER FILE NAME":PRINT:PRINT"     ";:INPUTF$
130 IFF$=""THENF$="NO NAME"

139 '"[illegible]
140 CLS:PRINT@67,"[illegible]  ";F$:PRINT
150 PRINT"     CATALOGUE":PRINT"     PAGE":PRINT"     LOAD"
160 PRINT"     SAVE":PRINT"     HARD COPY":PRINT"     NEW FILE"
170 PRINT"     ALPHABETICAL SORT":PRINT"     [illegible]"
190 A$=INKEY$:A$=""
200 A$=INKEY$:IFA$=""THEN200
210 IFA$="P"THEN390
220 IFA$="H"THEN520
230 IFA$="A"THEN850
240 IFA$="C"THEN280
250 IFA$="S"THEN1070
260 IFA$="L"THEN1000
270 IFA$="N"THENRUNELSE200
```

```
279 '"[illegible]
280 CLS:PRINT@3,"CATALOGUE  ";F$:PRINT
290 A=1:FORC=1TO40
292 IFZ$(A)="*"THEN350
294 IFZ$(A)="-"THEN310ELSEPRINT"PAGE"C;
300 PRINTZ$(A):GOSUB330
310 A=A+10
320 NEXT:GOSUB340:GOTO400
330 FORT=1TO400:NEXT:SOUND31,1:RETURN
340 A$=INKEY$:A$=""
350 PRINT"END OF CATALOGUE":PRINT"MENU  PAGE":SOUND28,2
360 A$=INKEY$:IFA$=""THEN360
370 IFA$="M"THEN140
380 IFA$="P"THEN390ELSE360


389 '"[illegible]
390 CLS:PRINT@230,"PAGE NUMBER";:INPUTP:IFP<1ORP>40THEN390
400 L=P*10-9:CLS
410 PRINT"PAGE "P;"  ";F$:PRINT
420 FORI=1TO10:PRINTUSING"###";L;:PRINT" ";Z$(L):L=L+1
430 NEXT:SOUND25,1
440 PRINT" [illegible] LINE INSERT":PRINT"        DELETE PAGE MENU"
450 A$=INKEY$:A$=""
460 A$=INKEY$:IFA$=""THEN460
470 IFA$="L"THEN690
480 IFA$="I"THEN720
490 IFA$="D"THEN780
500 IFA$="P"THEN390
510 IFA$="M"THEN140ELSE460
519 '"[illegible]


519 '"[illegible]
520 PRINT@418,"[illegible]"
530 A$=INKEY$:A$=""
540 PRINT@450,"[illegible] PAGE(S)  FILE  MENU":SOUND28,1
550 A$=INKEY$:IFA$=""THEN550
560 IFA$="F"THENP=1:F=40:GOTO610
570 IFA$="M"THEN140
580 IFA$="P"THEN590ELSE550
590 PRINT"  [illegible] START PAGE NUMBER";:INPUTP
600 PRINT"         FINAL PAGE NUMBER";:INPUTF:SOUND28,1
610 X=1:LPRINTF$:LPRINT
620 FORI=PTOF:LPRINT"PAGE"I
630 FORJ=XTOX+9:IFZ$(J)="*"THENLPRINT"  PAGE EMPTY":GOTO140
640 LPRINTUSING"##";J;:LPRINT" ";Z$(J)
650 NEXT
660 X=X+10
670 NEXT
680 LPRINT:GOTO140


689 '"[illegible]
690 PRINT@391,"LINE NUMBER";:INPUTL:IFL>400THEN690
700 FORD=417TO447:PRINT@D,CHR$(32):NEXT:SOUND30,1
710 PRINT@417,"[illegible]";:INPUTZ$(L):GOTO400

719 '"[illegible]
720 PRINT@391,"LINE NUMBER";:INPUTL
730 FORD=417TO477:PRINT@D,CHR$(32):NEXT
740 PRINT@417,"[illegible]";:INPUTN$
750 FORJ=-400TO-L:Z$(ABS(J))=Z$(ABS(J+1)):NEXT
760 Z$(L)=N$
770 GOSUB1600:GOTO400

779 '"[illegible]
780 PRINT@391,"LINE NUMBER";:INPUTL
790 X=400-L
800 FORJ=LTOX
810 Z$(J)=Z$(J+1)
820 NEXT
830 Z$(400)="-"
840 GOSUB1500:GOTO400
```

```
849 '"[illegible]
850 GOSUB960:PRINT@321,"[illegible] START LINE";:INPUTS:SOUND15,1
860 PRINT@353,"[illegible] FINAL LINE";:INPUTF:SOUND17,1:J=0:I=0
870 PRINT@416,"[illegible]":SOUND19,3
880 IFS<1ORS>400RF<SORF>400THEN970
890 FORJ=STOF:IFZ$(J)="-"THENZ$(J)="*"
900 K=F-S:FORI=1TOK
910 FORI=FTOSSTEP-1
920 IFZ$(I)<Z$(I-1)THENTP$=Z$(I):Z$(I)=Z$(I-1):Z$(I-1)=TP$
930 NEXT:NEXT:GOTO400

940 FORJ=1TO400:Z$(J)="-":NEXT:FORJ=1TO400STEP10:Z$(J)="*":NEXT
950 RETURN


960 CLS:PRINT@262,"[illegible]":RETURN

970 PRINT@416,"[illegible]":SOUND4,9
980 GOTO400


999 '"[illegible]
1000 PRINT@33,"[illegible]":COLOR4
1010 PRINT"[illegible]":SOUND28,9:CLS
1020 INPUT#"",F$:PRINTF$:PRINT
1030 FORI=1TO400
1040 INPUT#"",Z$(I):PRINT@448,"                "
1045 IFZ$(I)="*"THEN1060
1048 PRINTI;Z$(I)
1050 NEXT
1060 PRINT"[illegible]":SOUND30,9:GOTO140


1069 '"[illegible]
1070 CLS:PRINT@417,"[illegible]"
1090 A$=INKEY$:A$=""
1100 PRINT@450,"[illegible] FULL FILE OR [illegible] ONLY ":SOUND28,1
1110 A$=INKEY$:IFA$=""THEN1110
1120 IFA$="F"THENS=1:F=400:GOTO1180
1130 IFA$="P"THEN1150
1140 IFA$="M"THEN140ELSE1110
1150 PRINT"  [illegible] START PAGE NUMBER";:INPUTP
1160 PRINT"          FINAL PAGE NUMBER";:INPUTF
1170 S=P*10-9:F=F*10
1180 CLS:PRINT"[illegible]":SOUND28,9
1190 PRINT#"",F$:PRINTF$
1200 FORI=STOF
1210 PRINT#"",Z$(I)
1212 IFZ$(I)="*"THEN1230
1214 PRINTUSING"##";I;:PRINT" "Z$(I)
1220 NEXT
1230 PRINT"[illegible]":SOUND30,9:GOTO140


1500 G=INT(L/10)*10+10:IFZ$(G)="*"THENZ$(G)="-"
1505 IFZ$(G-9)="-"THENZ$(G-9)="*"
1510 FORI=GTO400STEP10:IFZ$(I)="*"THENZ$(I)="-"
1520 NEXT:G=G+1
1530 FORI=GTO391STEP10:IFZ$(I)="-"THENZ$(I)="*"
1540 NEXT
1550 RETURN


1600 G=INT(L/10)*10+11
1610 FORI=GTO400STEP10:IFZ$(I)="-"THENZ$(I)="*"
1620 NEXT:G=G+1
1630 FORI=GTO392STEP10:IFZ$(I)="*"THENZ$(I)="-"
1640 NEXT
1650 RETURN
```